

Learning & Personality Types: A Case Study of a Software Design Course

*Faheem Ahmed, Piers Campbell, Ahmad Jaffar,
and Shayma Alkobaisi*
*College of Information Technology,
United Arab Emirates University, Al Ain, UAE*

f.ahmed@uaeu.ac.ae; p.campbell@uaeu.ac.ae; ajaffar@uaeu.ac.ae;
shayma.alkobaisi@uaeu.ac.ae

Julie Campbell
*Faculty of Education & Language Studies,
The Open University, UK*

Julie.lynne.campbell@gmail.com

Executive Summary

The software industry has continued to grow over the past decade and there is now a need to provide education and hands-on training to students in various phases of software life cycle. Software design is one of the vital phases of the software development cycle. Psychological theories assert that not everybody is fit for all kind of tasks as people have different personality traits and abilities. The learning pattern of students is influenced by the personality types, with individuals having different personality types their learning pattern varies. The personality type of an individual generates a great deal of impact on the performance of various activities that humans can carry out. The behavior of individuals with a particular type of personality reflects the way these people perceive the world and make decisions. This personality type classification covers many aspects of human behavior, such as attitude, action and reaction, thinking, learning, feeling, and lifestyle. In this paper we provide empirical evidence about the significance of the personality types and discuss the learning pattern in an undergraduate software design course. The experiment was conducted on a subject of 85 students over the period of two years from 2007 to 2008. We employed the Myers-Briggs Type Indicator (MBTI) after carefully assessing other well regarded personality testing techniques such as MMPI (Minnesota Multiphasic Personality Inventory) and 16PF (Sixteen Personality Factor Questionnaire) to identify the personality type. MBTI establishes four parameters for assessing personality types. We all have the personality qualities

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact HPublisher@InformingScience.org to request redistribution permission.

contained within each scale or parameter, but we naturally prefer some qualities or are more comfortable with some styles than others. With the MBTI, each scale is bimodal with its central point having a zero value. Quantitative mechanisms were also employed in the form of paper based exams, assignments, projects, and quizzes for evaluating learning patterns of students. The results of the study found that individu-

als identified as ISTJ (using MBTI), that is students possessing the personality traits of Introvert, Sensing, Thinking, and Judging performed at a consistently higher level than any other group of students. However, it was also found that those students who were classified as ENTJ (Extrovert, Intuition, Thinking, and Judging) also demonstrated high levels of attainment. Clearly these two types possess the common traits of Thinking and Judging; however there is an obvious disparity in the other personality traits exhibited. The commonalities and differences exhibited by the two groups and their levels of attainment in the course obviously demonstrate that in the Software Engineering discipline a measured and methodological personality is more likely to perform well. In conclusion the findings of this experiment reinforce the current perceptions that personality types play a significant role in learning. It also provides empirical data about the personality type's involvement in teaching one of the core phases of software development life cycle.

Keywords: Software Engineering, Personality Types, MBTI, Software Design, Undergraduate Education, Human Factors, Performance Measures

Introduction

The Software Engineering curriculum in most of the universities around the world aims to provide education and hands-on training in various phases of the software development life cycle. The waterfall model of the software development cycle is generally covered in the curriculum of software engineering, despite being classified as an old approach to software development. Design is a fundamental activity in software development and involves the identification of the major sub-systems and their relationship at an abstract level. It involves problem solving and creating a workable and implementable solution. The software design is ultimately translated into code, which takes the physical shape of the software running on any machine. According to Budgen (2003) the design process in its classical form is a scientific approach to problem solving typically consisting of observing the characteristics of some phenomenon, making measurements of these, building a theory to explain them, and then seeking to verify these. According to Software Engineering Body of Knowledge (Hilburn, Hirmanpour, Khajenoori, Turner, & Qasem, 1999):

Software design area is concerned with the transformation of the statement of requirements into a description of how these requirements are to be implemented. Software design consists of activities such as architectural design, abstract specification, interface design, component design, data structure design, tasking design, and algorithm design. Software design uses a variety of techniques and forms of representation, each providing a capability for capturing and expressing a different view of the system. (p. 17)

The psychological hypothesis that not everyone can perform all tasks effectively reveals that personality traits or types play a critical role in the performance of people executing the same task. The personality type of an individual generates a great deal of impact on the performance of various activities that humans can carry out. The behavior of individuals with a particular type of personality reflects the way these people perceive the world and make decisions. This personality type classification covers many aspects of human behavior such as attitude, action and reaction, thinking, learning, feeling, and lifestyle. In this work we are concentrating on the role of personality type in learning one of the core phases of software development life cycle. The study will aid us in understanding how different personality types of the students influence progress in a software design course. Keeping in view the significantly important phases of the software development cycle, the objective of this study is to investigate the impact of the personality type in acquiring the knowledge and necessary skills for the design process. The overall objective of this study is to find an answer to the following research question (RQ):

RQ: Does personality type have an impact on the learning capability of the students in acquiring skills and knowledge of the software design process?

Literature Survey

There have been several studies that have dealt with the personality types, particularly in computer science and software engineering education. It is a common phenomenon that people have different personality traits, and the way they perceive, plan, and execute any assigned task is influenced by it. Software is a product of human activity, which often includes problem solving capabilities, cognitive aspects, and social interaction. On the other hand, human beings are more complicated and less predictable than computers. Therefore, the complexity of human personality gives rise to intricate dynamics along software development that cannot be ignored, but have often been overlooked. In their paper, McCaulley, MacDavid, and Walsh (1987) provide empirical evidence that the students having the personality traits of introvert and thinker are most likely to complete their degree programs, while students classified as extroverts and perceivers do not move along with their degree completion requirements. Felder and Silverman (1988) examined personality types with respect to performance in several introductory and advanced-level engineering classes and found that, in terms of course grades, introverts outperformed extroverts, intuitors typically outperformed sensors, thinkers outperformed feelers, and judges typically outperformed perceivers. In their paper, Thomas, Ratcliffe, Woodbury, and Jarman (2002) examined the learning styles of 107 introductory computer science students and found that there were slightly more active than reflective learners, a clear majority of visual learners over verbal, and equal proportions of sensing or intuitive and sequential or global learners. Thomas et al. (2002) also found that reflective learners typically outperform active learners and verbal learners outperform visual learners with respect to exam grades and course grades.

Several empirical studies have investigated the relationship between the Myers-Briggs Type Indicator (and its associated factor types **Extraversion**, **Sensing**, **Thinking**, **Judging**, **Introversion**, **iNtuition**, **Feeling** and **Perceiving**, which are discussed fully in the next), and software engineering. Sitton and Chmelir (1984) list some stereotypes of programmers and what it is that attracts them to the computer field. Their study paints a picture of creative professionals merrily and irreverently solving complicated problems, untrammelled by routine and humdrum details; however, it gives no specific statistics regarding their findings. Bush and Schkade (1985) tested 58 professionals involved with scientific programming in one aerospace company. They found that those possessing the factors of Introversion, Sensing, Thinking, and Judging (ISTJ 25%) to be the most common type, with the second most frequently reported type being INTJ (16%), and ENTP (9%) as the third. They also found thinking (74%) and judging (70%) to be very common. Buie (1988) took a sample of 47 scientific computer professionals employed by a private company under contract with NASA who were performing work on orbital-related software. ISTJ (19%), INTP (15%), and INTJ (13%) were the most frequent types, with those three collectively accounting for nearly half the sample. ESFJ (0%), ISFP (0%) and ENTP (0%) were particularly underrepresented. The hypothesis that scientific programmers would tend toward an over-representation of Is, Ns, and Ts was supported. Smith (1989) dealt with 37 systems analysts at a large insurance company. The most frequent types in the sample are ISTJ (35%) and ESTJ (30%). From the results, there were slightly more introverts (57%), but there was also a heavy bias towards the sensing (81%), thinking (89%), and judging (86%) types. Lyons (1985) surveyed 1229 software professionals from more than 100 companies, including insurance companies, financial institutions, utilities, and hardware manufacturers. He too found ISTJ (23%) to be the most common type, INTJ (15%) to be the second, and INTP (12%) to be a close third, noting that these three types composed 50% of his sample. He found thinking (81%) and judging (65%) types to be in the majority; furthermore, he also found that 67% of his subjects were classified as introversion types. Hardiman (1997) has claimed that the Myers-Briggs Type Indicator (MBTI) may be the best pre-

dictor of who will become a competent programmer. He observed that the majority of good software engineers were ENTJ, INTJ, ESTJ, ISTJ, ISFJ, and ENTP, in brief, mostly NTs and SJs. He implies that NF types tend to have trouble with the sequential and process-oriented thinking required to design and implement software.

Within the field of software engineering there is agreement that there are tremendous differences among individuals' achievement in programming. Instructors of programming courses witness firsthand the huge variety among students in learning achievement and programming assignments. Bishop-Clark (1995) investigated the relationship between cognitive aspects, personality traits, and computer programming. She divides programming in several stages: problem representation, program design, implementation, and debugging. She organized the theories and the empirical studies of computer programming into four sub-tasks: problem solving, designing, coding, and debugging. The cognitive styles discussed in some detail by Bishop-Clark include: field dependency/independency, analytic/holistic, impulsivity/reflectivity, and divergent thinking; the personality traits include focus of control and introversion/extroversion. These variables were mentioned because, according to her theory, they were all important within the realm of computer programming. Cognitive styles have been studied as factors that may help explain some of the variability; however, they have failed to consistently explain individual preference towards computer programming as opposed to, say, systems analysis. MBTI offers a potential to provide a suitable model for comparison. Capretz (2003) investigated the profile of a group of 100 software engineers (80% male and 20% female) who study in private or public universities or who work for the government or for software companies and found that NT and ST types were over-represented in the sample, while SF and NF were underrepresented.

Experimental Design & Setup

The target population for this study covers the fourth year undergraduate software engineering students from United Arab Emirates University. The experiment was conducted on a subject of 85 students over the period of two years from 2007 to 2008. The target population had already taken some basic programming and software engineering courses. We used MBTI personality assessment method, which serves as the measuring instrument in this study, to find the personality type of various students. We used quantitative mechanisms of paper based exams, assignments, and quizzes for evaluating learning pattern. The students were required to fill out the MBTI measuring tool and their personality characteristics were recorded. We followed the recommended ethical principles to ensure that the experiment conducted and reported here would not violate any form of recommended experimental ethics. The primary ethical principle in human subject research is that of full informed consent on the part of the subject to participate in the research project (Faden, Beauchamp, & King, 1986; Katz, 1972). We fully informed the participants in this study about the nature and objectives of this study. Moreover the information we acquired from respondents does not reflect the personal information of the individual. All the participants of this study were volunteers and no compensation in any form was offered or paid. As a starting point, it is fundamental to introduce the four scales of the MBTI (Myers, McCaulley, Quenk, & Hammer, 1998) which is used as the measuring tool in this study. The next sub-section provides some introductory information about the measuring tool of this study.

Measuring Tool

The human factor, which largely determines the individual characteristics of the personality, can be assessed using some of the many personality tests such as MMPI (Minnesota Multiphasic Personality Inventory), MBTI (Myers-Briggs Type Indicator), and 16PF (Sixteen Personality Factor Questionnaire). MMPI, developed originally in the early part of the 20th Century and revised by Butcher, Dahlstrom, Graham, Tellegen, & Kaemmer (1989), is frequently used to assess an indi-

vidual's mental health and is wide ranging in its coverage, asking participants questions related to a broad variety of issues from their perception of their body and health to their social interaction habits. The 16PF test was also developed in the early 20th century by Cattell (1946) and revised most recently in 1993 (Cattell, Cattell, & Cattell). 16PF uses sixteen categories to assess an individual's personality ranging from Warmth (attitudes to others) to Vigilance (is an individual trusting or suspicious). The test has been widely used in both the assessment of students and also potential employees, however the results of studies using 16PF have been mixed (Eysenck & Eysenck, 1985; Schuerger, Zarrella, & Hotz, 1989). In light of the type of research being conducted in this study and the mixed results of the 16PF test, MBTI was selected as the most suitable testing methodology. In addition, a number of studies in the area of software engineering and human factor have employed MBTI, which supports our selection of this testing technique for the personality assessment of students in the our experiment. MBTI is one of the most widely used personality assessments (Bishop-Clark & Wheeler, 1994; Devito, 1985; Edwards, Lanning, & Hooker, 2002; Smither, 1998). MBTI provides information about how people prefer to receive information, how they form opinions, and how they communicate. The MBTI ranks now among the most popular tools used in the workplace for analyzing personality types. This indicator establishes four parameters for assessing personality types. We all have the personality qualities contained within each scale or parameter, but we naturally prefer some qualities or are more comfortable with some styles than others. With the MBTI, each scale is bimodal with its central point having a zero value. Each respondent is asked to choose preferences; the higher the score on each preference, the stronger that preference is likely to be.

Extroversion and Introversion (E and I)

The first scale represents complementary attitudes towards the external world. As identified by Capretz and Ahmed (2010), an extrovert looks outward to the world and relishes engagements with other, while an introvert is less likely to engage with other and the outside world as a whole. For example, strong extroverts are sometimes said to “talk to think” whereas introverts “think to talk”. As Capretz and Ahmed point out, the “the implications of these terms go beyond the everyday caricatures of sociable versus shy.” Extroverts are talkative, initiators of conversation, and outgoing. They prefer action and variety. Introverts, in contrast, are quiet, respondent to conversation, and reserved. They like quiet and time to consider matters. Extroversion is described as a tendency to be talkative, lively, and expressive; introversion describes a person that is quiet, introspective, and reserved.

Sensing and Intuition (S and N)

The second scale deals with the how an individual absorbs and interprets information from the environment they are in. An individual who is described as sensing typically absorbs information which is presented to them and which is tangible. Individuals who are classed as Intuitive are more likely to rely on “gut feeling” and prior experience when interpreting new information. They will rely less on concrete tangible facts and more on the theoretical concepts related to the information or problem at hand. Capretz and Ahmed (2010) also suggest that there is a fundamental difference in the type of tasks that these two classes prefer to undertake. They suggest that sensing individuals like repetitive tasks which follow a defined series of steps, while those who rely on intuition prefer new tasks which are imagination driven and require more creative solutions. Capretz and Ahmed do however note that typically an individual will have some traits from each of these two groups; however one will be more dominant.

Thinking and Feeling (T and F)

The third scale of orientation in the MBTI classification is thinking and feeling. This scale governs how an individual reaches a decision having been given information relevant to a task. Those

who are *thinkers* exhibit more detachment during their reasoning process, examining the data from a structured viewpoint and deciding which is the most logical decision based on the facts. *Feelers*, on the other hand, examine the problem from a more personal point of view. They will examine the evidence from a more internal aspect and try to reach a decision which is the best fit for a given problem. Feelers typically also examine the affects of a decision on any external elements, which would not normally be considered by those classified as thinkers.

Judging and Perceiving (J and P)

The fourth scale differentiates between how we orient our lifestyles and organize our world. Capretz and Ahmed (2010) note that Judging identifies the tendency to be super organized, while Perceiving typically describes a person who appears to be disorganized and seems to be distracted from completing a task until the last minute. Myers (1995) supports this view stating that judging individuals like to settle matters early while those who are more perceiving prefer to leave key decisions open until the very end of a process. Capretz and Ahmed go on to identify a set of descriptors which can be used for judging types – deadlines, punctual, closure – and another for perceiving types – open-ended, tentative, adaptable, and spontaneous.

Quantitative Measuring Tool

We used quantitative mechanisms of paper based exams, assignments, project, and quizzes for evaluating learning patterns. The distribution of the marks for evaluating students is shown in Table 1, whereas Table 2 illustrates the grades distribution scheme. Five individual assignments were given to the students, and it was required that each student do these assignments by his or herself. Each assignment was due at the beginning of class on the date specified. Project based assessment is another strategy that can be used in the assessment of student learning where projects are assigned so a student can draw on his or her body of knowledge to solve a complex problem (Goda & Reynold, 2010). Therefore, students were required to work individually to create and develop a unique professional design of a software project. They were allowed to choose the topic and domain of the software project from a limited pool of project choices. They had to cover the assigned criteria and meet the deadlines. The objective of the project is to provide students an opportunity to explore areas and aspects of the software design of their particular interest and to channel their creativity and enthusiasm toward a productive and rewarding goal. A midterm and a final examination were required for the course; Table 3 illustrates the format and structure of the midterm and final examination. Each student was required to write the midterm and final exam individually. These examinations were closed book. Some spot quizzes were planned which were also closed book. In the course outline, the following course outcome targets were conveyed to the students:

- Explain the software design process and relate its importance in software development cycle.
- Identify, describe and list the software design notations and produce software design using notations.
- Outline software design quality fundamentals and issues related to design and prepare software design quality metrics.
- Differentiate software design methods and learns to indentify, relate, and select different design methods according to requirements and situations.
- Design, generate, and modify software patterns and their use.

Table 1: Quantitative Measuring Instrument

Final Exam	35 %
Midterm Exam	25 %
Assignments	15 %
Quiz	10 %
Project & Presentation	15 %
TOTAL	100 %

Table 2: Grades Distribution Scheme

Grade Title	Marks
A	90-100
B	85-89
B+	80-84
C+	75-79
C	70-74
D	60-69
F	<= 59

Table 3: Format of Mid and Final Term Examination

Question Type	No. of Questions in Mid Term Exam	No. of Questions in Final Exam
Multiple Choice	1	1
Descriptive short Answer	2	4
Descriptive Essay	1	2
Problem Solving & Activity	2	4
Total Questions	6	11
Total Marks	25	35

Teaching Methodology

The primary teaching technique used in this course was collaborative problem solving. The instructor first teaches specific contents to the class with hands-on examples to ensure the cognitive processing of the concepts in solving typical design issues. In the interactive lecture sessions the students were engaged in applying the concepts in class assignments and discuss and provide feedback to the instructor. Active learning, or learning by doing, has been used in academia and training for a long time; learning becomes active when students employ their creative skills during the learning process (Stankovic & Tillo, 2009). An active listening and communication envi-

ronment was encouraged in the class room by asking and clarifying questions and encouraging elaboration. We also used simulated scenario based laboratory environment where students were directed to carry out certain hands-on assignments to understand the broader theoretical concepts. According to Johnson, Lorents, Morgan, and Ozmun (2004) and Watson and Schneider (1999), such outcome is directed at cross appreciation of theories covered in lecture applied to a simulated scenario based laboratory environment. Drysdale, Ross, and Schulz (2001) demonstrate that people who choose terms such as 'analytical', 'logical', 'objective', 'ordered', 'persistent', 'product-oriented' and 'rational' to describe themselves tend to do well in mathematics, science, and technology.

The selection of the collaborative based problem solving pedagogical approach was motivated by the intention to observe how the different personality types of students affected their engagement within the class as a whole and in activities, in particular. This follows recent work in the education field by Lee, Nicoll, and Brooks (2004) and earlier studies by Christian and Titus (1998), which explored the role of collaborative problem based learning practices in the scientific field and found that such a pedagogical approach more realistically simulated future workplace conditions and permitted students freedom from obvious instructor supervision. As collaborative based problem solving requires key abilities in the areas of perception (understanding the problem and others views of the problem), the ability to communicate within a group setting and the ability to define the main issues and plan for a solution, the MBTI testing approach was selected as most suitable as it enabled students to be classified by a number of parameters which mapped directly to the abilities required under collaborative problem solving pedagogical approach. A number of the other measuring approaches offered similar opportunities, however, MTBI was found to be the most inclusive of the techniques considered. Moreover, during the learning of the software design process the students are required to carry out a wide range of tasks, which include prototyping, elaborating processing functions, and defining inputs and outputs. This shows that sensing and thinking characteristics are highly desirable for becoming software designers. Moreover during the problem solving the students are required to make key decisions which have direct and indirect impact on the design, therefore, judging characteristics also complement the design process.

Results & Analysis

The subject population of this study was required to fill out the questionnaire for personality types profiling. Table 4 illustrates the number of occurrences of the subjects in different personality characteristics. Each cell shows a personality trait in terms of four MBTI factors. Figure 1 illustrates the personality profile of the student data of this study. It is observed and reported here that a large number of students were belonging to ISTJ and ENTJ types. Figure 1 illustrates the distribution of the personality traits among students. ISTJ and ENTJ share the 28% and 18% of the dataset. This illustrates that T (Thinking) and J (Judging) covers the 65% (ISTJ, ENTJ, ESTJ, INTJ) of the population. The second largest group covers ISTJ and ISFJ with 28% and 9% respectively. This shows that I (Introversion) and S (Sensing) covers the 45% (ISTJ, ISFJ, ISFP, ISTP) of the dataset. The lowest representative of the personality traits of 2% each has a common dimension of E (Extroversion).

Table 4: Personality Traits Classification of the Study Data

ISTJ	ISFJ	INFJ	INTJ
24	8	2	6
ISTP	ISFP	INFP	INTP
3	3	4	4
ESTP	ESFP	ENFP	ENTP
3	3	2	2
ESTJ	ESFJ	ENFJ	ENTJ
2	2	2	15

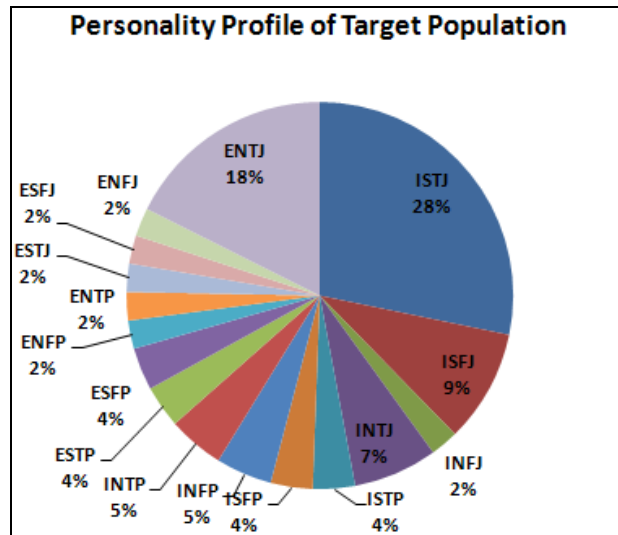


Figure 1: Personality Profile of the Student Data

The overall performance of the students with the personality traits classification is illustrated in Figure 2. Most of the ISTJ students received an “A” grade (about 50%), whereas 33% received “B+”, and 17% received “B”. None of the ISTJ type student received “C” or lower. In the case of ENTJ types, 40% of the students received “A”, 27% received “B+”, and 20% received “B” grades. In case of ENTJ, 13% of the students also received “C+”, but none of the students received a grade lower than “C+”. One student each from INTJ and INFP also received “A”. The students from types ESFP, ENFP, and ENTP received the lowest grade of “D”.

The performance of the students in spot quizzes with the personality traits classification is illustrated in Figure 3. Most of the ISTJ students received an “A” (about 67%), whereas 25% received “B+”, and 8% received “B”. None of the ISTJ type student received “C” or lower. In the case of ENTJ types, 53% of the student received “A”, 27% received “B+”, and 13% received “B”. In the case of ENTJ, 7% students also received “C+” but none of the student received a grade lower than “C+”. In case of ISFJ, none of the student received “A” but 50% received “B+” and 25% received “B” and One student each from INTJ and INFP also receives “A” grade. The students from types ESFP, ENFP, and ENTP received the lowest grade of “D”.

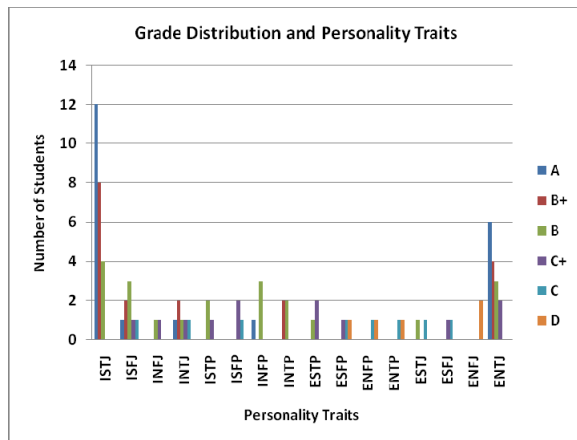


Figure 2: Overall Grade Distribution and Personality Traits

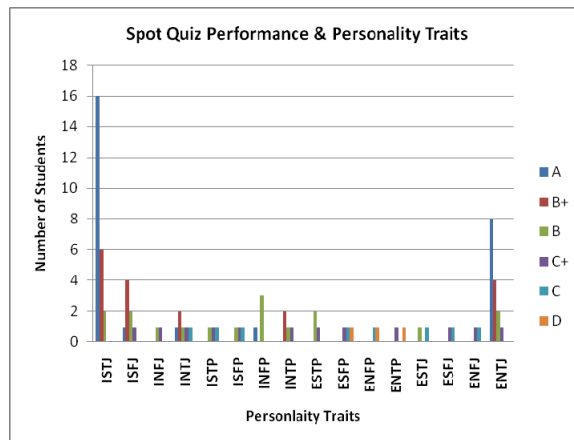


Figure 3: Spot Quiz Performance and Personality Traits

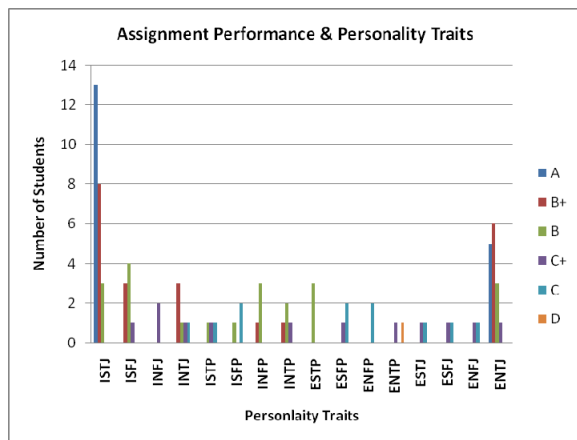


Figure 4: Assignment Performance and Personality Traits

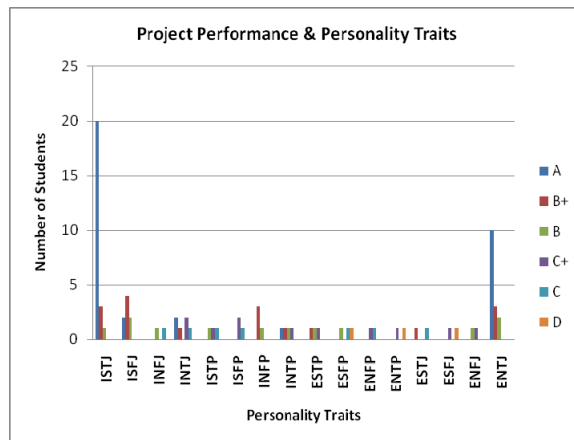


Figure 5: Project Performance and Personality Traits

The grades of the students in assignments with the personality traits classification are illustrated in Figure 4. Most of the ISTJ students received an “A” (about 54%), whereas 33% received “B+”, and 13% received “B”. None of the ISTJ type student received “C” or lower. In the case of ENTJ types, 33% of the student received “A”, 40% received “B+”, and 20% received “B”. In the case of ENTJ, 7% students also received “C+” but none of the students received a grade lower than “C+”. In the case of ISFJ, none of the student received “A” but 37% received “B+” and 50% received “B”, whereas 13% received “C+”. One student of ENTP type received the lowest grade of “D”.

The performance of the students in the individual project with the personality traits classification is illustrated in Figure 5. Most of the ISTJ students received “A” (about 83%), whereas 13% received “B+”, and 4% received “B”. None of the ISTJ type student received “C” or lower. In the

case of ENTJ types, 67% of the student received “A”, 20% received “B+”, and 13% received “B”. None of the ENTJ type student received “C+” or lower. In the case of ISFJ, 25% of the student received “A”, 50% received “B+” and 25% received “B”. One student from INTP also received “A”. The students from types ESFJ, ENFP, and ENTP received the lowest grade of “D”.

Limitations & Threats to External Validity

Surveys, experiments, metrics, case studies, and field studies are examples of empirical methods used to investigate both software engineering processes and products (Singer & Vinson, 2002). Such experiments are subject to certain limitations, which is the case in this study as well. Threats to external validity are conditions that limit the researcher’s ability to generalize the results of his/her experiment to industrial practice (Wohlin et al, 2000), which was the case with this study. One of the main limitations and threats to external validity is localization of the dataset. The entire participant sample in this experiment belongs to the same geographical location (United Arab Emirates) and university. The instructor of the course who quantitatively evaluated the students was the same for all of the students. Another limitation of this study is the small sample size. Although the experiment covers a dataset of 85 students, which is arguably reasonable, we believe that a larger size would increase external validity. Although the proposed approach has some potential to threaten external validity, we followed appropriate research procedures by conducting and reporting tests to improve the reliability and validity of the study, and certain measures were also taken to ensure the external validity.

Discussion

The software industry generally complains that current university curricula fail to address the practical issues of real software development (Coulter & Dammann, 1994; Dawson, Newsham, & Kerridge, 1992; Gibbs, 1989; Shaw, 1990; Wasserman, 1996). Design holds a central position in all software engineering activities, either directly or indirectly; thus, it is essential that software engineering education provides solid and in-depth instruction in design (Freeman, Wasserman, & Fairley, 1976). Software designers have a wide range of tasks, which include prototyping, elaborating processing functions, and defining inputs and outputs. Although there is huge diversity among design principles, we can find common principles that are applicable to the design of any artefact, whether it is a poster, a household appliance, or a housing development. The first part of the design stage may require characteristics similar to those required for analysis, as it involves team discussions and interaction with the user. Although software design is still a young field and far from having a consensus on its relevant principles, software design will always require the human creativity possessed by other disciplines, such as architecture, marketing, or graphic design, rather than the hard-edge formulaic certainty of engineering design. Software design is an exploratory process, the designer looks for components by trying out a variety of schemes in order to discover the most natural and reasonable way to refine the solution.

In his 2003 study, Capretz found more introverts (I=57%) than extraverts (E=43%), significantly more sensing (S=67%) than intuitive (N=33%) types, greatly more thinking (T=81%) than feeling (F=19%) types, and more judging (J=58%) compared to perceiving (P=42%) types. The results of this study show that those who possess the traits of thinkers and judges (who we will refer to as judges for the purposes of this work) perform significantly better than students who possess other personality traits, such as perception. As shown in Figures 2-5 individuals identified as ISTJ, that is students possessing the personality traits of Introvert, Sensing, Thinking, and Judging performed at a consistently higher level than any other group of students. It is also important to note that another group which showed high levels of attainment is the ENTJ personality group (Extrovert, Intuition, Thinking and Judging). Clearly these two types possess the common traits of Thinking and Judging; however there is a clear disparity in the other personality traits exhibited.

The commonalities and differences exhibited by the two groups and their levels of attainment in the course obviously demonstrate that in the Software Engineering discipline a measured and methodological personality is more likely to perform well. For example, one of the design activities is use case modeling (Armour & Miller, 2001), which provides techniques for identifying an appropriate system boundary, understanding the interactions of external entities with the system, developing the capabilities the system should provide, and understanding the domain the software problem is situated in. The definition of the use case modeling clearly requires thinking and judging.

This inference is perhaps somewhat at odds with common perceptions, in which it is assumed that software engineering is a more creative branch of engineering in which the more imaginative personality traits (such as intuition, perception, and feeling) might be more beneficial than the methodological approaches. However if we examine the results more closely, we can see that in course Assignments (Figure 4) students who possessed the more creative traits of perception, feeling, and intuition gained higher marks in this assessment type than in Spot Quizzes (Figure 3) and projects (Figure 5). We suggest that this improvement in performance is linked to their ability to be more creative given the freedom of their practical assignments. This is in stark contrast to their performance in more constrained assessment exercises, such as examinations and quizzes, and indicates that when given more freedom students with more creative personality traits are able to perform more successfully.

Software designers should have the ability to see the big picture and the ability to single out the items that are relevant from imprecise and large quantities of fuzzy data, which requires the ability to find patterns. Our results demonstrate that students who possess a more organized methodological personality typically perform to a higher level in the academic setting than those who are less methodological yet more creative and intuitive in their personal make up and approach to problem solving and academic study. The results further demonstrate that personality type is a critical factor in academic performance and that in an engineering or scientific domain an organized methodological approach yields significantly higher results. It is interesting to note that of the four factor groups identified by Myers Briggs Type Indicator, only half of these factors have a significant impact on student performance, namely Thinking and Judging. It is also interesting to note that students who are more introverted performed better than their more extroverted colleagues, indicating perhaps that these students are more insular and less focused on personal interactions.

We would stress that our results do not necessarily transfer to the industrial setting in which creativity, feeling, perception, and an outgoing personality might be an advantage. Indeed our results could suggest that those students who perform best are best suited to solitary employment, rather than team working; however, these issues are outside the boundaries of the current study which focuses solely on academic performance. The results of the study may also have been affected by cultural factors as mentioned earlier (Limitations & Threats to External Validity), in that the learning culture in the United Arab Emirates is a more traditional approach in which students focus on developing standard methodological techniques that can be reproduced in order to solve a given problem type. This pedagogical system is similar to the rote learning techniques applied in western countries in the recent past, which enabled the repetitive solving of problems but which discouraged creative approaches and alternative techniques. There is a need to better prepare graduates beyond the theoretical concept by incorporating industrial operational experiences equipped with the application of advance technology within a simulated classroom environment (Joseph & George, 2002). Keeping in view this mind, we are working on industrial partnerships to improve the teaching and learning methodology of this course. Its effectiveness can be further optimized by incorporating a direct partnership with local industries that are using similar technology. While a generic industrial internship is a practice within higher educational institutions,

such industrial interaction and collaboration should be based on an objective specific course learning outcome and help in further improving the effectiveness of the learning outcomes.

Conclusion

Better software will result from the combined efforts of a variety of mental processes, outlooks, and values. In this study our objective was to understand the role of the personality types in student learning in one of the most important phases of a software development life cycle course. The results of this investigation reinforce the current perceptions that different personality traits are most suited for different tasks. In the case of software design, the students having the characteristics of Judging and Thinking show better performance in comparison to other personality traits. The use of interactive technologies may provide us with a valuable guide for designing technologies where the individual learner may find him or herself in an environment that both instructs about the subject matter and encourages the learner to construct knowledge from subject matter in more meaningful and effective way (Bendar, Cunningham, Duffy, & Perry, 1992). Specifically, utilizing state-of-the art technologies that reflect current industry practices not only enhances students' confidence in their ability to successfully perform as an Information Systems major, but also elevates students' expectations that valued rewards will be received by majoring in IS (Akbulut & Looney, 2009). Although this work has certain limitations, the work contributes to a better understanding the learning pattern of future software designers.

References

- Akbulut, A. Y., & Looney, C. A. (2009). Improving IS student enrollments: Understanding the effects of IT sophistication in introductory IS courses. *Journal of Information Technology Education, 8*, 87-100. Retrieved from <http://www.jite.org/documents/Vol8/JITEv8p087-100Akbulut297.pdf>
- Armour, F., & Miller, G. (2001). *Advanced use case modeling: Software systems*. Addison-Wesley.
- Bendar, A. K., Cunningham, D., Duffy, T. M., & Perry, J. D. (1992). Theory into practice: How do we link? In T.M. Duffy & D. H. Jonassen (Eds.), *Constructivism and the technology of instruction: A conversation*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Bishop-Clark, C. (1995). Cognitive style, personality, and computer programming. *Computers in Human Behavior, 11*(2) 241-260.
- Bishop-Clark, C., & Wheeler, D. (1994). The Myers-Briggs personality type and its relationship to computer programming. *Journal of Research on Computing Education, 26*(3), 358-370.
- Budgen, D. (2003). *Software design*. Addison Wesley.
- Buie, E. A. (1988). Psychological type and job satisfaction in scientific computer professionals. *Journal of Psychological Type, 15*, 50-53.
- Bush, C. M., & Schkade, L. L. (1985). In search of the perfect programmer. *Datamation, 31*(6), 128-132.
- Butcher, J. N., Dahlstrom, W. G., Graham, J. R., Tellegen, A., & Kaemmer, B. (1989). *The Minnesota Multiphasic Personality Inventory-2 (MMPI-2): Manual for administration and scoring*. Minneapolis, MN: University of Minnesota Press.
- Capretz, L. F. (2003). Personality types in software engineering. *International Journal of Human-Computer Studies, 58*(2), 207-214.
- Capretz, L. F., & Ahmed, F. (2010). Making sense of software development and personality types. *IEEE IT Professional, 12*(1), 6-13.
- Cattell, R. B. (1946). *The description and measurement of personality*. New York: World Book.
- Cattell, R. B., Cattell, A. K., & Cattell, H. E. P. (1993). *16PF fifth edition questionnaire*. Champaign, IL: Institute for Personality and Ability Testing.

Learning & Personality Types

- Christian, W., & Titus, A. (1998). Developing web-based curricula using Java physlets. *Computers in Physics*, 12, 227-232.
- Coulter, N., & Dammann, J. (1994). Current practices, culture changes, and software engineering education. *Computer Science Education*, 5(2), 211-227.
- Dawson, R. J., Newsham, R. W., & Kerridge, R. S. (1992). Introducing new software engineering graduates to the 'real world' at the GPT company. *Software Engineering Journal*, 7(3), 171-176.
- Devito, A. J. (1985). Review of Myers-Briggs Type Indicator. In J. Mitchell (Ed.), *The ninth mental measurements yearbook* (pp. 1030-1032). Lincoln, Nebraska.
- Drysdale, M. T. B., Ross, J. L., & Schulz, R. A. (2001). Cognitive learning styles and academic performance in 19 first-year university courses: Successful students versus students at risk. *Journal of Education for Students Placed at Risk*, 6(3), 271-289.
- Edwards, J. A., Lanning, K., & Hooker, K. (2002). The MBTI and social information processing: An incremental validity study. *Journal of Personality Assessment*, 78(3), 432-450.
- Eysenck, H. J., & Eysenck, M. W. (1985). *Personality and individual differences: A natural science approach*. New York: Plenum.
- Faden, R. R., Beauchamp, T. L., & King, N. M. P. (1986). *A history and theory of informed consent*. Oxford University Press.
- Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering Education*, 78, 674-681.
- Freeman, P., Wasserman, A. I., & Fairley, R. E. (1976). Essential elements of software engineering education. *Proceedings of the 2nd International Conference on Software Engineering*, pp. 116-122.
- Gibbs, N. (1989). The SEI education program: The challenge of teaching future software engineers. *Communications of the ACM*, 32(5), 594-605.
- Goda, B. S., & Reynold, C. (2010) Improving outcome assessment in information technology program accreditation. *Journal of Information Technology Education*, 9, IIP 49-59. Retrieved from <http://www.jite.org/documents/Vol9/JITEv9IIPp049-059Goda769.pdf>
- Hardiman, L. T. (1997). Personality types and software engineers. *Computer*, 30(10), 10.
- Hilburn, T. B., Hirmanpour, I., Khajenoori, S., Turner, R., & Qasem, A. (1999). *A software engineering body of knowledge Version 1.0*. TECHNICAL REPORT CMU/SEI-99-TR-004 ESC-TR-99-004.
- Joseph, G., & George, A. (2002). ERP, learning communities and curriculum integration. *Journal of Information Systems Education*, 13(1), 51-58.
- Johnson, T., Lorents, A. C., Morgan, J., & Ozmun, J. (2004). A customized ERP/SAP model for business curriculum integration. *Journal of Information Systems Education*, 15(3), 245-253.
- Katz, J. (1972). *Experimentation with human beings*. New York: Russell Sage Foundation.
- Lee, K. M., Nicoll, G., & Brooks, D. W. (2004). A comparison of inquiry and worked example web-based instruction using physlets. *Journal of Science Education and Technology*, 13(1), 81-88.
- Lyons, M. L. (1985). The DP psyche. *Datamation*, 31(16), 103-110.
- McCaulley, M. H., MacDavid, G. P., & Walsh, R. (1987). Myers-Briggs Type Indicator and retention in engineering. *International Journal of Applied Engineering Education*, 3, 99-109
- Myers, I. B., with Myers, P. B. (1995). *Gifts differing: Understanding personality type*. Mountain View, CA: Davies-Black Publishing.
- Myers, I. B., McCaulley, M. H., Quenk, N. L., & Hammer, A. L. (1998). *MBTI Manual: A guide to the development and use of the Myers-Briggs Type Indicator*. Consulting Psychologists Press.

- Schuerger, J. M., Zarella, K. L., & Hotz, A. S. (1989). Factors that influence the temporal stability of personality by questionnaire. *Journal of Personality and Social Psychology*, 56, 777-783.
- Shaw, M. (1990). Prospects for an engineering discipline of software. *IEEE Software*, 7(6), 15-24.
- Singer, J., & Vinson, N. G. (2002). Ethical issues in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 28(12), 1171-1180.
- Sitton, S., & Chmelir, G. (1984). The intuitive computer programmer. *Datamation*, 30(20), 137-140.
- Smith, D. C. (1989). The personality of the systems analyst: An investigation. *ACM Computer Personnel*, 12(2), 12-14.
- Smither, R. D. (1998). *The psychology of work and human performance* (3rd ed.). New York: Longman.
- Stankovic, N., & Tillo, T. (2009). Concurrent software engineering project. *Journal of Information Technology Education*, 8, IIP 27-41. Retrieved from <http://www.jite.org/documents/Vol8/JITEv8IIP027-041Stankovic715.pdf>
- Thomas, L., Ratcliffe, M., Woodbury, J., & Jarman, E. (2002). Learning styles and performance in the introductory programming sequence. *Proceedings of SIGCSE '02*, pp. 33-37.
- Watson, E. E., & Schneider, H. (1999). Using ERP systems in education. *Communications of the Association for Information Systems*, 1, 1-47.
- Wasserman, A. I. (1996). Toward a discipline of software engineering. *IEEE Software*, 13(6), 23-31.
- Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., & Wesslen, A. (2000). *Experimentation in software engineering*. Norwell, MA: Kluwer Academic Publishers.

Biographies



Faheem Ahmed received his MS (2004) and Ph.D. (2007) in Electrical Engineering from the University of Western Ontario, London, Canada. Currently he is assistant professor at UAE University, United Arab Emirates. Ahmed had many years of industrial experience and during his professional career he has been involved in the life cycle of software development process. Ahmed has authored and co-authored many peer-reviewed research articles in leading journals and conference proceedings in the area of software engineering.



Piers Campbell received his BSc (Hons), MRes and PhD degrees from the University of Ulster, Northern Ireland in 2000, 2001, and 2005 respectively. Piers is currently an assistant professor in the Faculty of Information Technology at the United Arab Emirates University, UAE. Prior to this he held a lectureship in the School of Computing and Information Engineering at the University of Ulster. Piers is the author or co-author of over 30 many peer-reviewed research articles.



With extensive experience in various IT positions, **Dr. Ahmad Jaffar** amassed a wealth of knowledge in business process design and improvement engineering and strategic business to IS/IT alignment. Prior to joining the United Arab Emirates University in 2003, he was a Computer Science Lecturer with the University of Wales, UK. His current research is in the areas of semiotics, cognitive psychology in human interpretation of representation as well as HCI in relation to implied symbolic representation.



Dr. Shayma Alkobaisi is a UAE national and currently an Assistant Professor at the College of Information Technology in the United Arab Emirates University. She received her Ph.D. degree in Computer Science from the University of Denver in June 2008. In 2009, she served as a member of the advisory council to the Science and Technology (S&T) program at Emirates Foundation and in 2010 she was recruited as Manager of the same program.



Julie Campbell received her BSc (2007) and M.Ed (2010) degrees from the Open University, UK. Julie has over 10 years of experience in the training and education sector spanning primary, secondary, tertiary and professional levels. Julie is currently the Director of Studies for Edgewater Middle East based in the United Arab Emirates. Edgewater Middle East is an overseas branch of Edgewater College, Ireland.